Magistrate Judge Laurel Beeler
U.S. District Court Northern District of California
Courtroom C - 15th Floor
450 Golden Gate Avenue
San Francisco, CA 94102

Re: *Uniloc USA, Inc. et al. v. Apple Inc.*
4:18-cv-00362-PJH (LB)

Dear Magistrate Judge Beeler:

Pursuant to this Court's Order Denying Motion to Compel Without Prejudice (Dkt. 130), the parties submit this joint letter brief to request this Court to resolve the dispute Uniloc raised in the previously filed motion to compel, ECF No. 124, which this Court denied, without prejudice. Lead counsel have met in person, in an unsuccessful attempt to resolve this issue.

Respectfully,

Clayton C. James (Cal. Bar No. 287800)
HOGAN LOVELLS US LLP
4085 Campbell Avenue, Suite 100
Menlo Park, CA 94025
Telephone: (650) 463-4000
Facsimile: (650) 463-4199

James J. Foster
PRINCE LOBEL TYE LLP
One International Place, Suite 3700
Boston, MA 02110
Telephone: (617) 456-8000
Facsimile: (617) 456-8100

Uniloc's position. The unresolved issue relates to Apple's source code production.

Relationship of this dispute to other Uniloc-Apple actions. There are 10 actions pending in this District in which Uniloc has sued Apple for patent infringement. These actions were transferred from the Eastern District of Texas in January 2018, and then randomly assigned. Because various of the actions were related, many were reassigned, leaving Judge Hamilton with three of the actions, Judge Alsup with six, and Judge Koh with one.

Because all 10 cases involve patents that are software-related, issues of infringement will likely turn on how Apple's source code controls the operation of the accused products. Uniloc thus needs to inspect that code to confirm the products infringe and to obtain evidence of that infringement to present to the Court.

Apple elected to produce its source code on a standalone computer for review as to all ten actions at the same location, without presorting the code by action or patent. It was left to William C. Easttom II, Uniloc's source code reviewer for all ten actions, to examine whatever code was on the computer and try to find the portions relevant to the functionality of the claims of the asserted patents.

As Easttom avers in the attached declaration, during the first several days of his code inspection, he discovered that large sections of code that he believed were relevant to his assignment were not on the computer. He called this to the attention of Uniloc counsel, who requested Apple to produce the code, which it has not done.

Because Apple did not presort the code by action or patent, this dispute as to code production is not limited to this action, but extends across all ten actions. Although this dispute was first presented to Judge Hamilton, specifically in the -362 action, and referred to Your Honor, the same dispute will be brought to the attention of Judges Alsup and Koh, and they will be informed of Your Honor's involvement in attempting to resolve the issue.

Judge Alsup has scheduled a case management conference in his six actions for April 26 at 8 AM. Judge Hamilton has also scheduled a case management conference in her three actions for April 26, but at 2 PM. Judge Koh has scheduled a case management conference for May 2 at 2 PM in the single case assigned to her. Uniloc plans to bring up this code production issue at each of those conferences, and report your involvement.

The dispute in this action. The '556 patent involved in this action is software-related. Proving infringement requires that Uniloc's source code expert, William C. Easttom II, review whatever source code he needs to review to be able to testify as to how the accused Apple products work. As he states in the attached declaration, he has been retained by various parties in litigation to review source code of over 100 companies that produce software, including the software of Microsoft, Symantec, and McAfee, but had never run into the difficulties encountered here. In several previous cases, where he has reviewed Apple software, Apple was fully cooperative. But again, not here.

Easttom journeyed to the offices of Apple counsel in Chicago to review the Apple code, as relevant to all ten actions, on a standalone computer, and spent six days there. During the first several days, he discovered that large sections of the code he believed relevant were not on the computer. Uniloc has since made repeated requests to Apple to produce that code for inspection, but Apple refused. As detailed in the attached declaration, Easttom drafted several memos to be forwarded to Apple counsel to identify the relevance of, and to assist them in identifying, the

missing code, as well as specific claim elements the code was related to, and Easttom's rationale for believing the missing code was relevant and important. Despite that, Apple still refused to produce the missing code.

This is not the way code review usually works. Typically, a code reviewer wants to inspect the code as efficiently as possible, confirm that he has reviewed all code he believes relevant, and get out. Similarly, the producing party normally wants to get the code review over as quickly and painlessly as possible, so to expedite the process it allows the reviewer to look at whatever he thinks relevant. Because there is invariably a rather strict protective order in place, misuse or abuse of the process is not a factor. As Easttom avers (¶7), when he previously inspected Apple source code in other litigation, Apple cooperated in this process.

The parties cannot go to Court every time they disagree – whether in good faith or bad – over whether particular portions of code could be relevant to this action, or the other nine actions. That would not only squander Uniloc's resources, but would burden the Court with the dreary task of repeatedly sorting through descriptions of source code, if not the source code itself. So Uniloc asked Easttom to draw on his experience and suggest the best way to proceed, in order to minimize his time and effort in reviewing the code, to reduce the burden to Apple, and at the same time to reduce, if not eliminate, the back-and-forth between counsel arguing about the details of code production.

He recommended (¶21-22), as a matter of convenience and efficiency to all, including Apple, that Apple simply produce the entire iOS and watchOS code for inspection, onsite at Apple if it preferred. As he outlined, that approach had no downsides. But, short of that, what is essential to Uniloc is that Easttom, as the code expert assigned to testify under oath as to how the products work, have access to the code that *he* believes governs how the products work. Apple cannot be the arbiter of what he gets to see, because that would allow Apple to conceal from his review portions of the code that demonstrate infringement.

The dispute as to this case illustrates the kind of games that can be played. The patent is directed to the use of motion detected by an accelerometer to measure vertical movement, for certain purposes. Easttom is seeking to review Apple source code that would reveal Apple uses an accelerometer to measure vertical movement, and identify the purposes to which the measured data is put.

Apple's stratagem has been to produce code related to a single Apple product feature called "Flights Climbed," and refuse to produce other code, claiming, as it does in this letter brief, the action is limited to that single feature. But Uniloc has informed Apple this action is not limited to the "Flights Climbed" feature, and Uniloc is currently accusing Apple of infringement beyond that feature (see Easttom Decl, ¶19). Easttom has discovered, from his review of the limited code made available to him, that Apple is using the motion detected by an accelerometer to measure vertical movement in other product features. ¶19. Apple is trying to prevent him from documenting that infringement. (Even the "Flights Climbed" production is incomplete, (¶18-19.) and Easttom needs to review the missing code to be able to understand and to testify about that feature.) Easttom must be allowed to review any source code that performs the infringing functionality.

Uniloc has requested, as an alternative to production of the entire iOS and watchOS code, that the Court schedule a Discovery Conference, and order that it be attended by an Apple representative with detailed familiarity of the code, as well as by Easttom. Uniloc believes that a

discussion between the Court and those two highly technically trained individuals will inform the Court of the problems that source code review presents, and lead to a solution. As far as scheduling, counsel for the parties will be attending the initial Case Management Conference in this case on April 26, in Judge Hamilton's courtroom, in Oakland. A discovery conference, if the Court orders it, could thus conveniently be scheduled for either April 25 or 27.

As to Uniloc's final proposed compromise, Uniloc had suggested an informal meeting between "code guys." Apple would bring someone with detailed familiarity of its code to meet with Easttom, and answer questions about the code that had not been produced. The meeting would be unrecorded, and any statements by the Apple representative would be unsworn, and off the record. Apple does not even have to disclose the representative's name or position. If Apple's "code guy" (or woman) persuades Easttom the missing code is not relevant, then it need not be produced. But otherwise, Easttom needs to inspect the code, to do his job.

Apple's Position. Fed. R. Civ. P. 26(b)(1) limits discovery to only that which is both relevant to the specific claims at issue and proportional to the needs of the case as defined by those claims. Rule 26(b)(2)(c) states that courts "must" limit discovery that exceeds the scope of 26(b)(1). Uniloc's demand for unfettered access to the entirety of Apple's iOS and watchOS source code is the antithesis of "proportional." It is both untethered to Uniloc's actual infringement contentions, which accuse a single Apple product feature called "Flights Climbed," and massively overbroad. Apple has been a party to over 700 patent litigation cases, and in all that history it has never produced anything close to the entire source code tree for any of its operating systems. Indeed, Uniloc requests access to Apple source code that is greater, by far, than Apple's own employees are allowed. For example, particular source code projects are made accessible only to the engineers who work on those or related projects, by employing user access restrictions, passwords, and other security protocols. (Kateley Decl. at ¶¶8-9.) Uniloc's demands also encompass confidential code owned by third-parties or that protects important third-party rights (id. at ¶¶9-10), such as the 2.8 million lines of FairPlay software that manages access rights to digital content. The code requested includes thousands of features and functions that have no relevance to this case.

The court should reject Uniloc's demand for "the entire iOS and watchOS code" and its other proposed solutions to this self-created dispute. Courts in this District stress that source code requests must be narrowly tailored and limited even where a protective order is in place, given the inherent sensitivity and risk of harm. *See In re Google Litig.*, 2011 WL 286173, at *4 (N.D. Cal. Jan. 27, 2011) (limiting demand for entirety of its source code to specifically accused feature because of "the potential harm from the disclosure of a firm's proprietary source code, even with the safeguards offered by a protective order."); *Nazomi Commc'ns, Inc*, 2012 WL 1980807, at *3 (rejecting plaintiff's request for complete source code "in order to fully understand the operation of Samsung's products" where Plaintiff has not demonstrated the necessity of full understanding versus understanding the portion that is covered by its infringement claims."); *Waymo LLC v. Uber Techs., Inc.*, 2017 WL 6883929, at *2 (N.D. Cal. Oct. 19, 2017) (denying plaintiff's motion to compel defendants' source code because it was "profoundly overbroad" and "untethered" to the plaintiffs' actual claims in the case at issue).

Source code is unquestionably among Apple's most precious and vulnerable assets—reflecting the work of thousands of Apple engineers and billions of dollars of investment. Its unauthorized disclosure would be potentially catastrophic to Apple and its customers, and Apple should not be forced to subject its extremely valuable source code to the serious risks

entailed in over-disclosure, particularly when the vast majority of the code demanded has no relevance to any issue at stake in this proceeding. For example, a source code leak could affect the security of Apple products, allowing bad actors to create security vulnerabilities. (Kateley Decl. at ¶7.) The concern of a source code leak in spite of a protective order is not unfounded, hypothetical, or even limited to misuse or inadvertence by Uniloc itself. *See, e.g. Valencell, Inc. v. Apple Inc.*, No. 5:16-cv-00001-D (W.D.N.C., Aug. 28, 2017) (addressing a "manifestly suspicious circumstance presented of two different FedEx shipments by the same law firm containing [source code] not reaching their proper destination").

The prejudice to Apple is exacerbated by Uniloc using the same source code reviewer in each of the ten cases transferred to this District and in the context of Uniloc suing Apple 22 times in the last several months. If Apple is required to produce to Uniloc every line of its source code, irrespective of whether it is related to any functionality accused in this case, it would only be providing fodder for Uniloc to buy additional patents to assert against Apple in this protracted campaign, particularly given lead counsel has accompanied Mr. Easttom during multiple days of source code review. Uniloc's purported assurances that Mr. Easttom will not look at source code that is not relevant to this case are unavailing. First, such misuse of source code may have already happened. Uniloc initially reviewed Apple source code related to updating software on devices, then purchased a different patent on controlling software upgrades and promptly sued Apple on that patent. Also, Apple is not comforted by Uniloc's representations about what it would or would not do if it had access to Apple's entire source code. This is particularly true given the backdrop of this litigation, where Judge Gilstrap previously devoted seven pages to documenting Uniloc misrepresentations before concluding they were "troubling" and "not isolated." *Uniloc USA, Inc. et al. v. Apple Inc.*, No. 2*:17-cv-00258-JRG* (E.D. Tex., Dec. 22, 2017.)

If Mr. Easttom and Uniloc were genuinely interested in presenting this Court with a manageable dispute, they would have worked with Apple to narrow their requests, including by identifying the actual claim limitations that allegedly require additional code and some explanation as to why the code that has been produced is insufficient for them to understand the relevant operations of the single feature at issue here. Instead, immediately before filing its motion to compel, Uniloc presented Apple with a disjointed memo from Mr. Easttom purporting to document source code that had been "withheld." As with the first third of Uniloc's letter, most of Mr. Easttom's complaints were directed to patents that are not part of this case, are at issue in other cases that are pending before other judges and who have not been presented with Uniloc's complaints. The current dispute is not a proper vehicle for addressing unbriefed complaints in those actions. Uniloc's only source code complaint directed at the three Uniloc cases currently before Judge Hamilton relates to the '556 patent, which claims a system for calculating an incline, based on using an accelerometer to measure a portions of human steps traveling up, traveling down and then calculating a difference between the two. The only accused functionality is Apple's "Flights Climbed" feature, which does not perform the '556 patent's step calculation but, instead, estimates flights ascended by measuring changes in air pressure. Uniloc does not deny that Flights Climbed is the only feature currently accused, but it apparently intends to try to accuse other product features in the future. On that basis, Uniloc asserts a need for Mr. Easttom to understand everything there is to know about how Apple uses accelerometers. For the only feature currently accused, however, Uniloc does not deny that Apple has produced all the

code Mr. Easttom needs to understand its operations in relation to the '556 patent's claims to calculating an incline using a particular algorithm, which Apple does not do. Nor does Uniloc suggest that the claimed algorithm might be implemented outside of the source code they have spent several days reviewing.

Therefore, this is not a dispute over "withheld" or "missing" code. Rather, Mr. Easttom has not found what he is looking for because it does not exist. Uniloc does not deny that Apple has produced all of the source code implementing the aspects of the Flights Climbed feature that have any connection to the '556 patent. Specifically, Apple's counsel supervised the collection and production of source code relevant to the accused features, and hereby attests that to date Apple has made the approximately 26,624 files and 6,393,551 lines of that source code available for Uniloc's inspection. Apple has produced all of the source code projects identified in the attached Declaration of Stephen Jackson as necessary for Mr. Easttom to confirm how the accused Flights Climbed feature operates. Finally, Apple has offered to produce the handful of source code files that Mr. Easttom has specifically identified, including the few files of "I/O Kit" software he has identified, though Apple does not believe they are relevant.

Apple's Proposal. The parties' impasse results from a combination of Uniloc's insistence that its source code reviewer must be the final arbiter of any dispute regarding whether requested source code is relevant and proportional to Uniloc's needs, coupled with its stated position that its source code needs are not bounded by its infringement contentions. That is not consistent with the law in this District or with the proportionality requirements of Rule 26. Instead, Apple has suggested a simple protocol for identifying any additional source code that Uniloc actually requires to understand and confirm the relevant operations of the accused features.

For the reasons above, the Court can and should summarily deny Uniloc's unbounded and unreasonable demand for additional source code. Uniloc's alternative request for a discussion among "code guys" is not workable here, given that there is no one "code guy" at Apple with the "detailed familiarity" of all iOS and watchOS code. Uniloc's insistence that if the parties and "code guys" do not agree, then Mr. Easttom's view must prevail, is also not workable. Similarly, as Uniloc refuses in meet and confers to even discuss any specific code that it wishes Apple to produce, there is no reason to think that a discovery conference or evidentiary hearing would be manageable or productive, particularly given that Uniloc contends that this Court should address disputes regarding "other actions" even though it acknowledges those other actions are before other judges before whom no dispute has been raised let alone referred to Your Honor for resolution. Instead, in order to focus whatever demands Uniloc is making, the Court should order Uniloc to comply with the process suggested by Apple and provide a chart listing the specific claim limitation for which it believes it needs additional source code, identifying the specific source code it needs and why, and explaining why the code already produced is insufficient with respect to that claim limitation. If relevant source code is identified, Apple will provide it for inspection. If the parties disagree about relevance or the sufficiency of particular aspects of Apple's production, then the procedure suggested by Apple will at least narrow the dispute and allow it to be presented to the Court in a more targeted fashion, subject to the full briefing this issue merits given the consequences to Apple of Uniloc's demands.